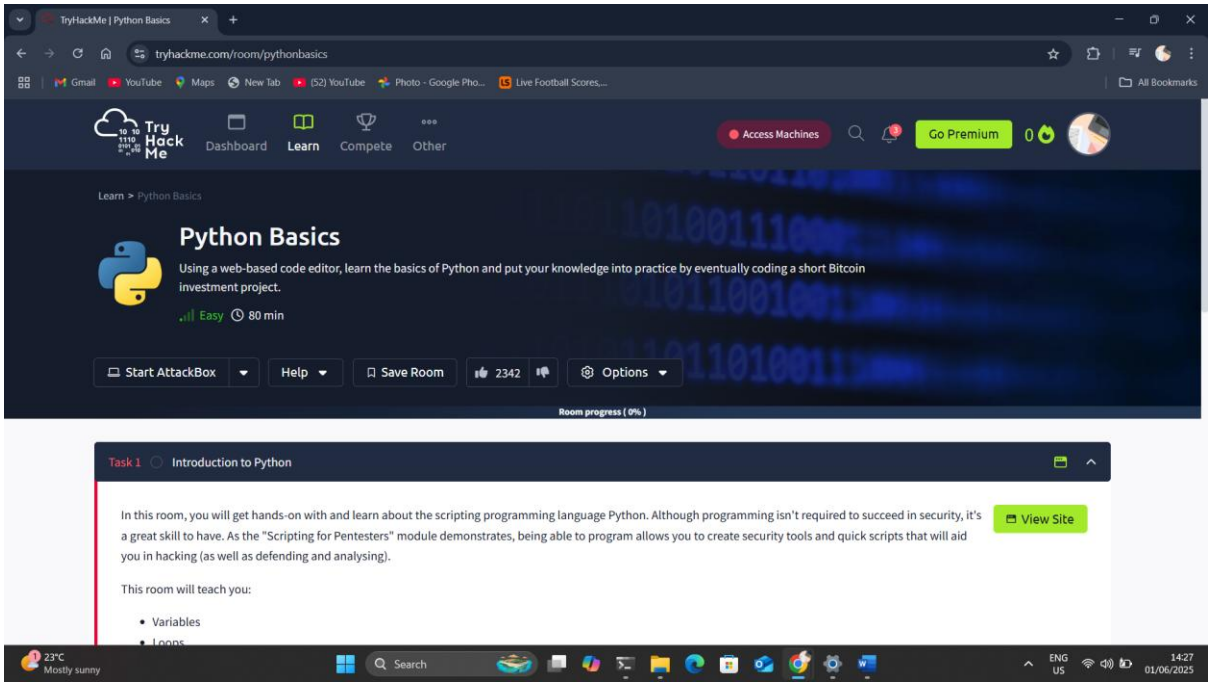
NAME:	**GRAHAM DESCENT OYIGO**
GMAIL:	**grahamoyigo19@gmail.com**
ADMISSION NUMBER:	**cs-sa10-25023**
MODULE:	**PYTHON BASICS ON TRYHACKME**
COMPLETION LINK:	https://tryhackme.com/p/grahamoyigo19?show_achievem

PYTHON BASICS ON TRYHACKME

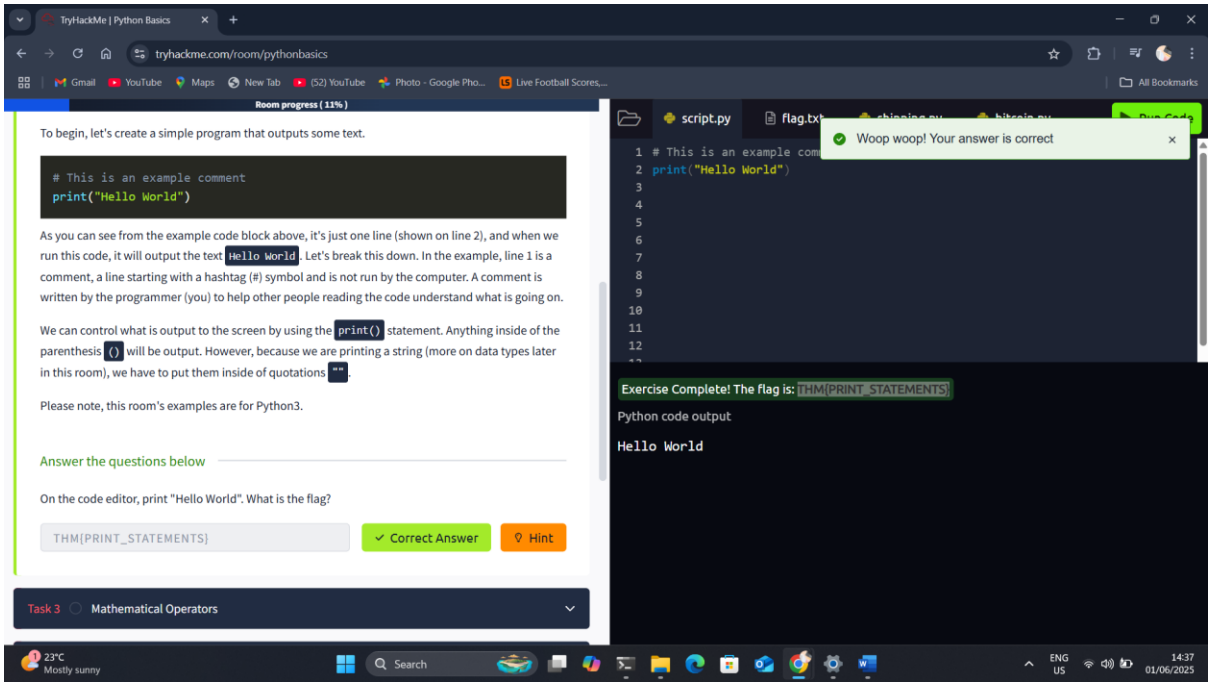
1. INTRODUCTION

I began this module by learning that In programming, syntax is important as it describes the structure of a valid programming language. In simple terms, syntax tells the computer how to read code using rules that control the structure of symbols, punctuation, and words of a programming language.



2. HELLO WORLD

Here, I learned how to output text in Python using the print() statement. I found out that when I want to display text, I need to put it inside parentheses () and enclose it in quotation marks (""). I also discovered that lines starting with a hashtag (#) are called comments; I can use them to add notes to my code, and the computer simply ignores them.



3. MATHEMATICAL OPERATORS

From this section, I learned about two important types of operators in Python:

First, I learned about **mathematical operators**, which allow me to perform calculations just like a calculator. I now know how to use:

- + for addition
    - for subtraction
- for multiplication
- / for division
- % for modulus (to get the remainder of a division)
- ** for exponentiation

Second, I learned about **comparison operators**, which are used to evaluate conditions. I understand these are crucial for future concepts like loops and if statements. The comparison operators I learned are:

> for greater than

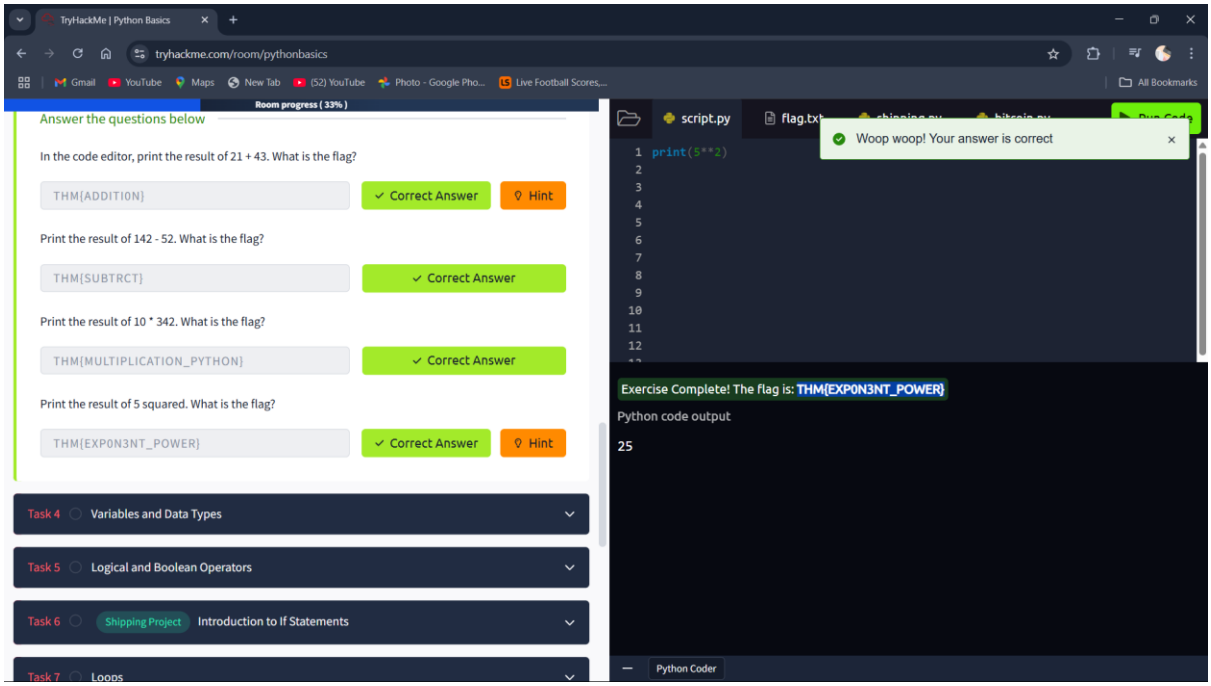
< for less than

== for equal to

!= for not equal to

>= for greater than or equal to

<= for less than or equal to

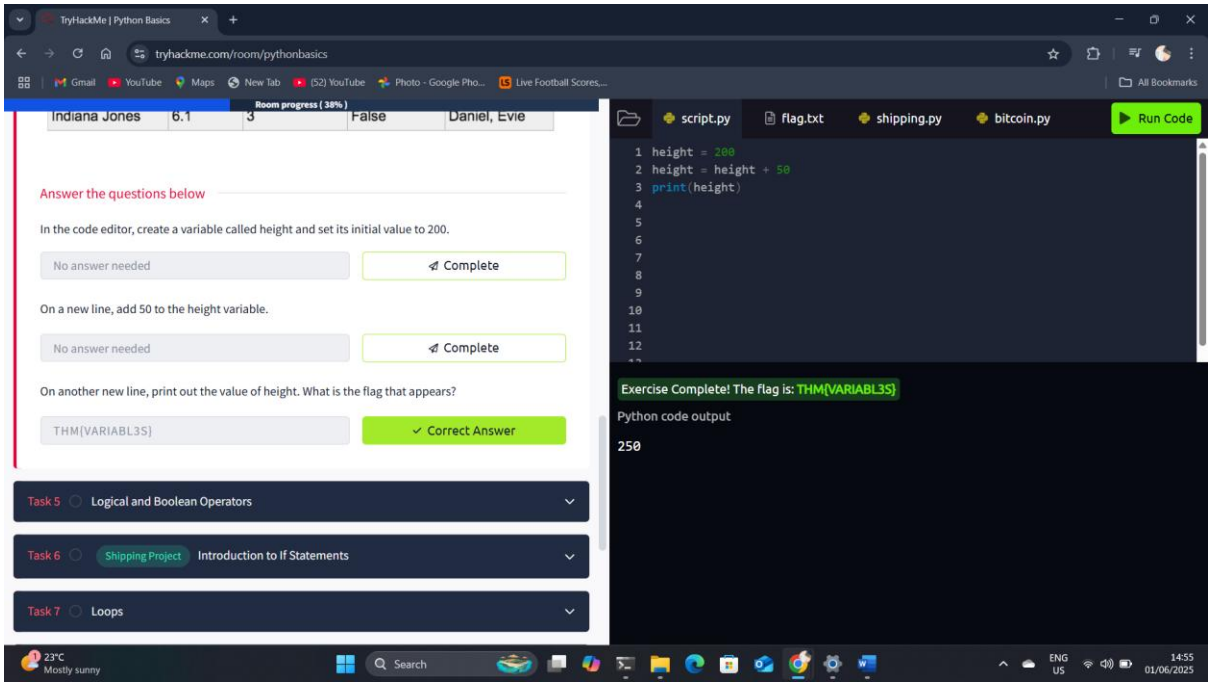


4. VARIABLES AND DATA TYPES

I learned that variables are used to store and update data in a program, giving a name to the data I want to keep track of (like food = "ice cream"). I also saw how to update a variable's value, for example, by increasing its current value.

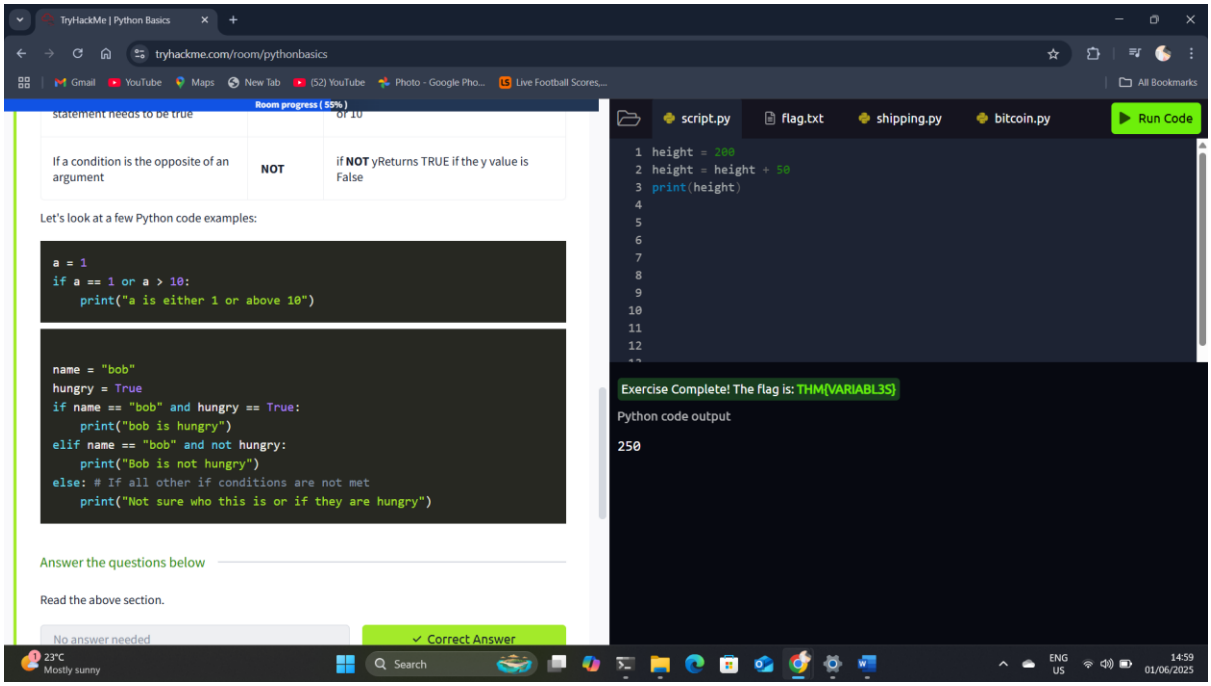
Regarding data types, I discovered that different kinds of data are stored in variables, and I learned about:

- String: for text (like "height")
- Integer: for whole numbers (like 2000)
- Float: for numbers with decimal points
- Boolean: for True or False values
- List: for collections of different data types



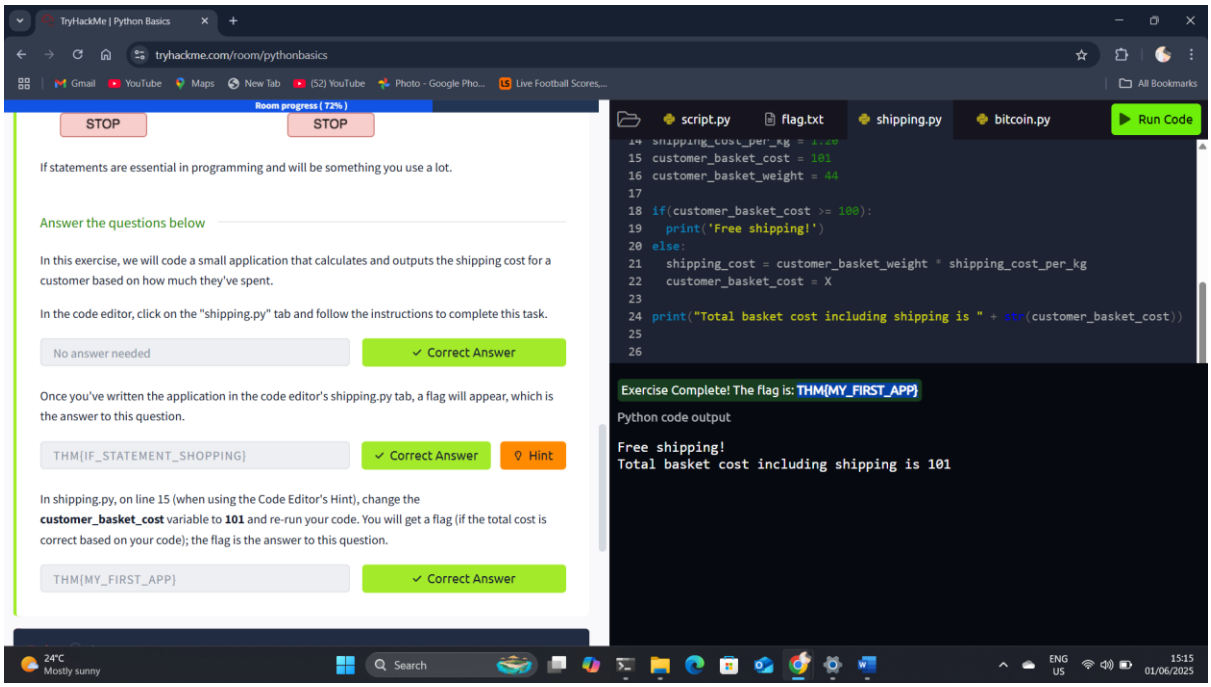
5. LOGICAL AND BOOLEAN OPERATORS

I learned that logical operators are essential for making comparisons and setting conditions in my code, especially within if statements. I can use operators like == (equal to), < (less than), <= (less than or equal to), > (greater than), and >= (greater than or equal to) to check relationships between values. Additionally, I discovered Boolean operators (AND, OR, NOT), which allow me to combine or modify conditions. AND means both conditions must be true, OR means at least one condition must be true, and NOT reverses a condition's truthfulness. These operators are vital for controlling how my program behaves based on different true or false scenarios.



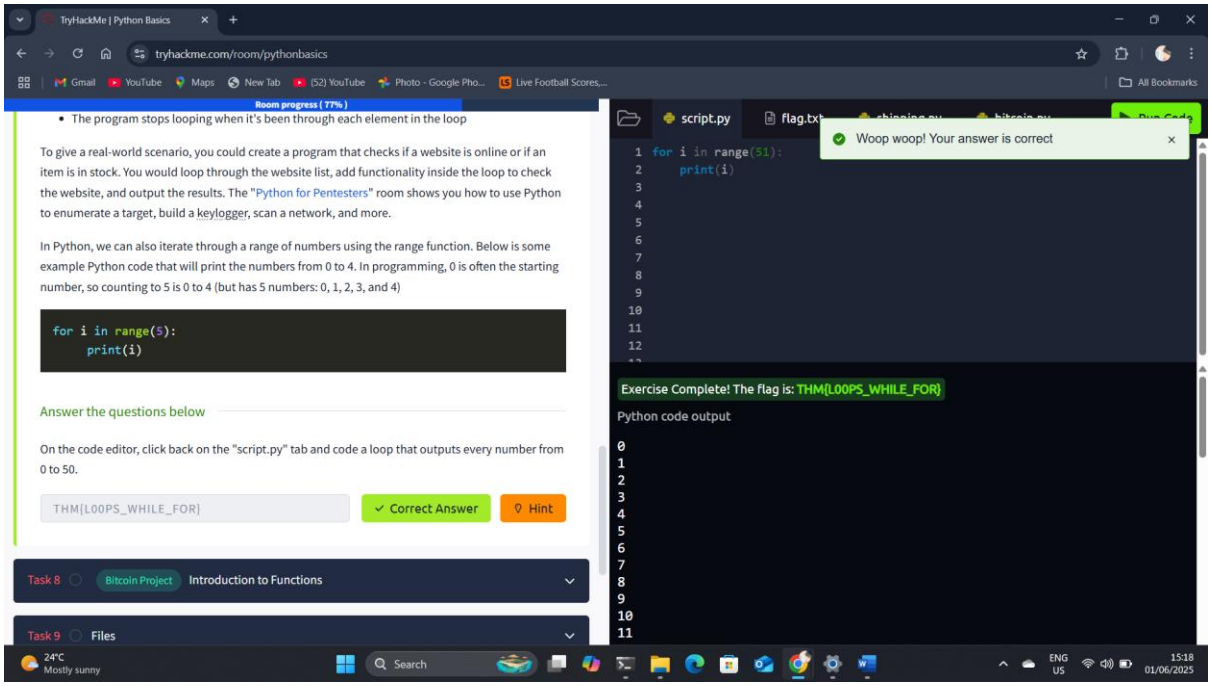
6. INTRODUCTION TO IF STATEMENT

In this task, I learnt that using "if statements" allows programs to make decisions. They let a program make a decision based on a condition. The if keyword indicates the beginning of the if statement, followed by a set of conditions. A colon : marks the end of the if statement. Anything after the colon that is indented, is considered part of the if statement, which the program will execute.



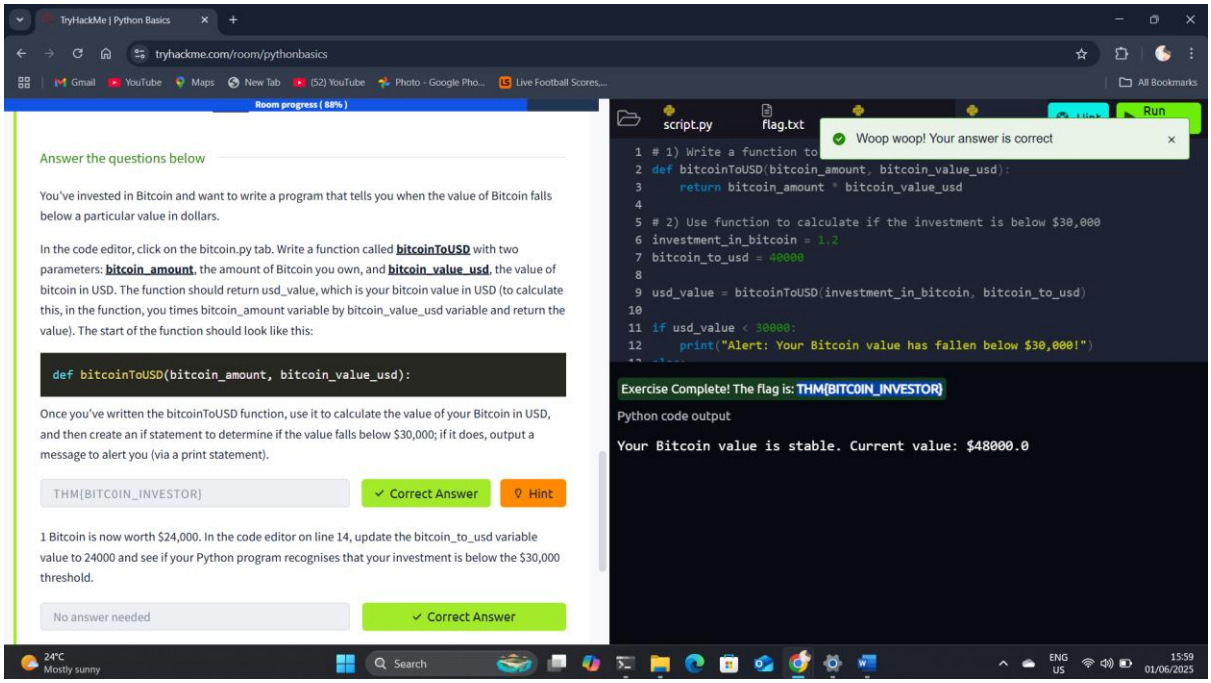
7. LOOPS

I learned that loops are fundamental for repeating actions in programming, and Python offers two main types: while and for loops. I understood that a while loop continues to run as long as a specified condition remains true, making it useful for iterating a set number of times or indefinitely until a condition is met. On the other hand, for loops are designed to iterate over sequences, such as items within a list, or even a range of numbers using the range() function, executing a block of code for each element in the sequence.



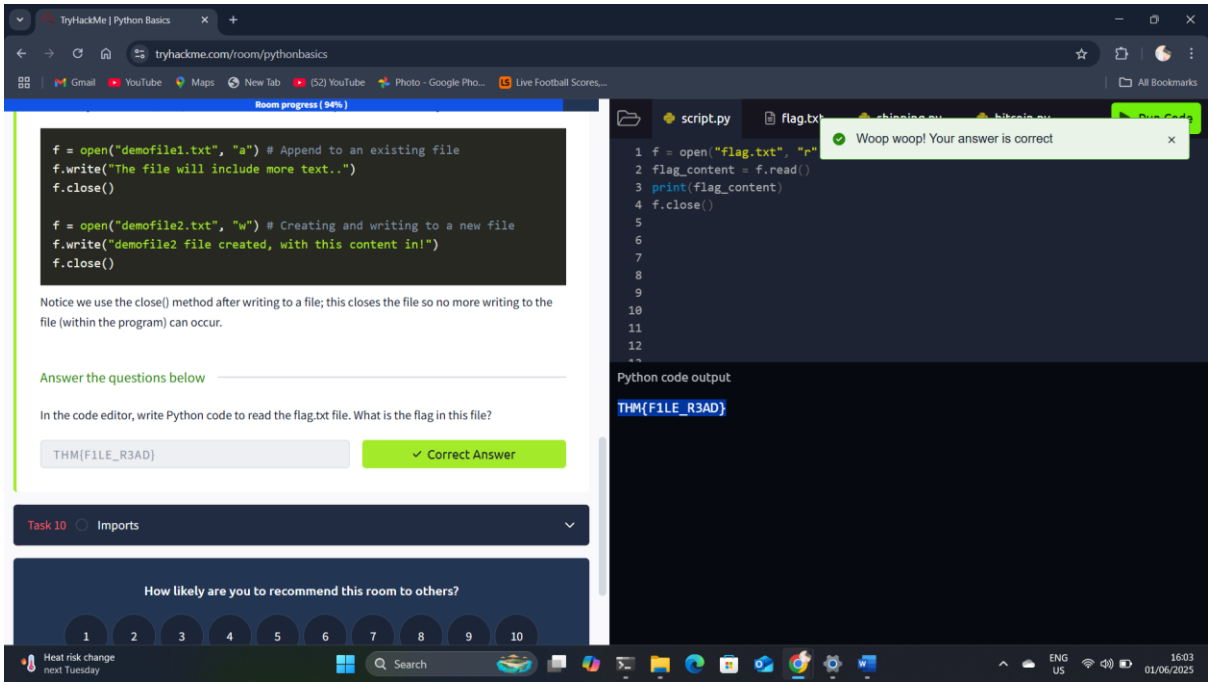
8. **INTRODUCTION TO FUNCTIONS**

Here, learned that **functions** are reusable blocks of code that help organize programs, especially as they grow larger and more complex. They allow me to avoid writing repetitive code by defining a set of instructions once and then calling that function whenever I need to perform those actions. I now know how to define a function using the def keyword, give it a name, specify **parameters** (input values) within parentheses, and use a colon and **indentation** to mark its body. I also understand that functions can **return** a result, which can then be used in other parts of my program.



9. FILES

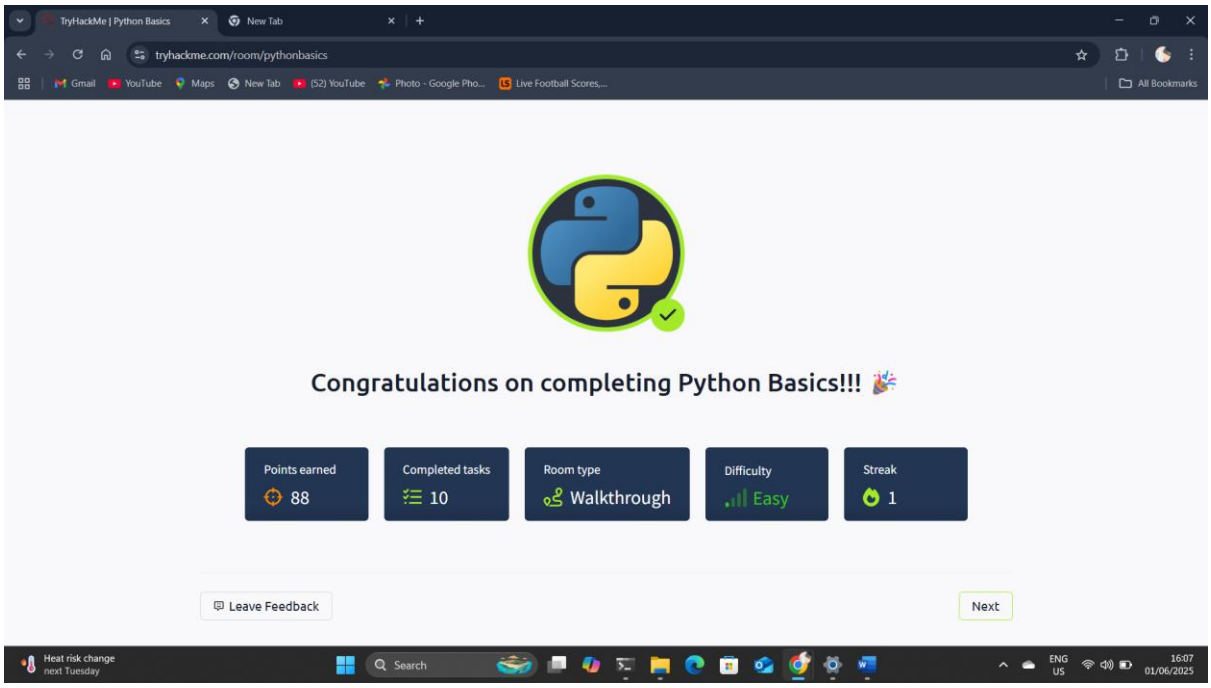
In Python, I learned that you can read and write from files. As seen in cyber security, it's common to write a script and import or export it from a file; whether that be as a way to store the output of your script or to import a list of 100's of websites from a file to enumerate.



I opened the file flag.txt using the open() function in read mode ("r") and used the read() method to retrieve its contents. After reading the data, I displayed it using print() to see the flag inside the file. Finally, I closed the file with the close() method to ensure it was properly handled.

10. IMPORTS

I learned that Python allows importing libraries, which are collections of prewritten functions that extend Python's capabilities. For example, the datetime library can be imported to work with dates and times, and its datetime.now() function gives the current date and time. Importing is done using the import keyword, followed by referencing functions using library_name.function_name(). Additionally, Python has a package manager called pip to install external libraries like scapy for crafting network packets, making Python highly extensible for tasks like pentesting.



11. CONCLUSION

Completing the Python basics on TryHackMe has provided me with a strong foundation in programming, including working with variables, functions, file handling, and libraries. I've learned how to write efficient code, utilize built-in methods, and import external libraries to extend functionality. Understanding these concepts has enhanced my ability to create scripts for various applications, such as automating tasks and analyzing data, which are essential skills in cybersecurity and beyond. This journey has prepared me to tackle more advanced Python concepts and apply them in real-world scenarios.